INTUIT mailchimp

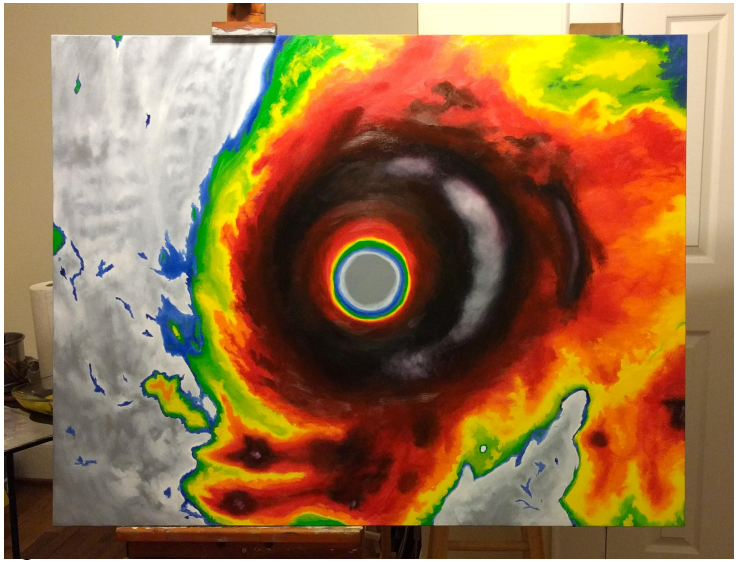
Extinguishing The Garbage Fire of ML Testing

Data Council Austin, 2023

Emily May Curtin
Staff ML Eng/Ops/Eng @ Intuit Mailchimp

Howdy, I'm Emily

- 👽 ATLien (don't call it Hotlanta)
- ❌ #NotADataScientist
- 🎨 Oil painter by passion
- 💾 MLEng/Ops/Eng by day job
- ❤️ Big fan of Ryan Curtin
- ⌨️ Contributor to FizzBuzzEnterpriseEdition



```java
package com.seriouscompany.business.java.fizzbuzz.packagenamingpackage.interfaces.factories;

import com.seriouscompany.business.java.fizzbuzz.packagenamingpackage.interfaces.strategies.FizzBuzzSolutionStrategy;

/**
 * Factory for FizzBuzzSolutionStrategy
 */
public interface FizzBuzzSolutionStrategyFactory {

        /**
         * @return
         */
        public FizzBuzzSolutionStrategy createFizzBuzzSolutionStrategy();

}
```


20
Catalyst
BY PRINCETON
20
Catalyst


20
Catalyst
BY PRINCETON
20
Catalyst

Thank you for validating my thesis by attending this talk.

ML Testing is a garbage fire

```
${PYTHON_BIN_PATH}/py.test -v --cov-report term-missing --cov-report xml:coverage.xml \
    --cov=script --cov=my_datascience_model_package --cov-config=tests/config/pytest_cov.cfg \
    integration_tests/ tests/

PATH=$PYTHON_BIN_PATH:$PATH tests/script/check_coverage_and_quality.sh $(pwd)/coverage.xml
```

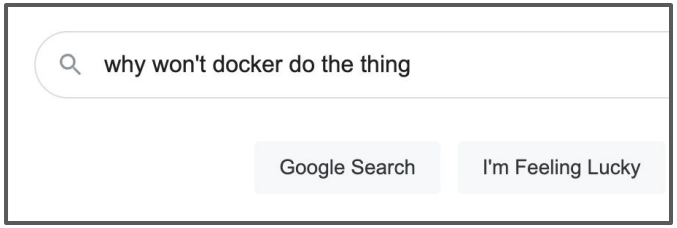
- It's difficult, time consuming, and clunky
- It's hard to get people to do it at all
- It's hard to get people to do it correctly
- Ultimately, applications don't work

GNU's Not Unix (v1.2)

by Daniel Kelly

C G C
Well GNU's Not Unix,
F C
What is Unix? You may ask,
F C
It's a Computer Operating System,
F G
From the distant past.
F C
Long before Micro was Soft, or
F C
Apple had a mouse,
F C
Unix was the solid rock,
C G C
On which Software built its house.

But Unix was a jealous beast,
Held tight by AT&T,
They'd send trade-secret binaries,
No source code you could see.
The price, it was a pittance,
But it cost your liberty:



How I Learned About Testing

1. Write an implementation of some data structure
2. Test the heck out of every little thing cause the auto-grader's gonna getcha
3. Fail all your own tests and find out that you did everything wrong
4. Rewrite and fix until you pass your own tests (or run out of time)
5. Hit "submit" with fear and trepidation
6. Find out from the auto-grader (unit test suite written by the TAs) that you did everything wrong
7. Cry
8. Rinse and repeat for more data structures and basic algorithms

We as engineers blow off steam by whining about Data Science code.

Taka
@astrobassball

don't touch that, it's my emotional support untitled jupyter notebook (12)

Data Scientists don't want to ship crap

Data Scientists don't want to ship crap.
So why does it keep happening?

Testing for ML Applications is...

I. ...a Technical Problem

II. ...a Human Problem

III. ...a different technical problem if you frame it right

IV. ...a Kumbaya opportunity, your key success metric, rah rah rah

A Technical Problem

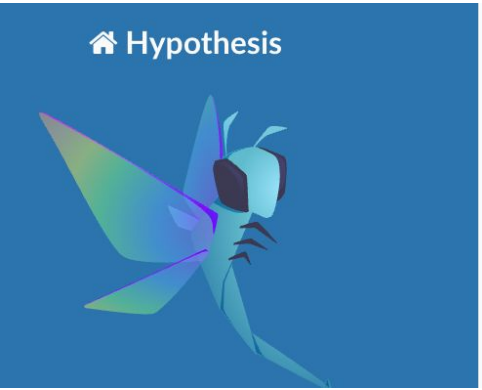
```python
import random

class MyModelIsSoAmazing:
    def predict(self, prediction_input):
        return random.uniform(0, 1)

```

Testing Probabilistic Code

- `hypothesis`
- `flaky`
- `approx()` handling for values




```python
from hypothesis import example, given, strategies as st


@given(st.text())
@example("")
def test_decode_inverts_encode(s):
    assert decode(encode(s)) == s
```

This can be useful to show other developers (or your future self) what kinds of data are valid

Ok cool but what about…

- Integrating into massive systems
- Dealing with prod or prod-like data
- Environment issues such as differences in hardware (ex: GPU)
- Retraining
- Using pytest in a notebook

A Human Problem

Code review and other good software engineering practices might make deployments less error-prone. However, because ML is so experimental in nature, they can be significant barriers to velocity; thus, many model developers ignore these practices (P1, P6, P11). P6 said:

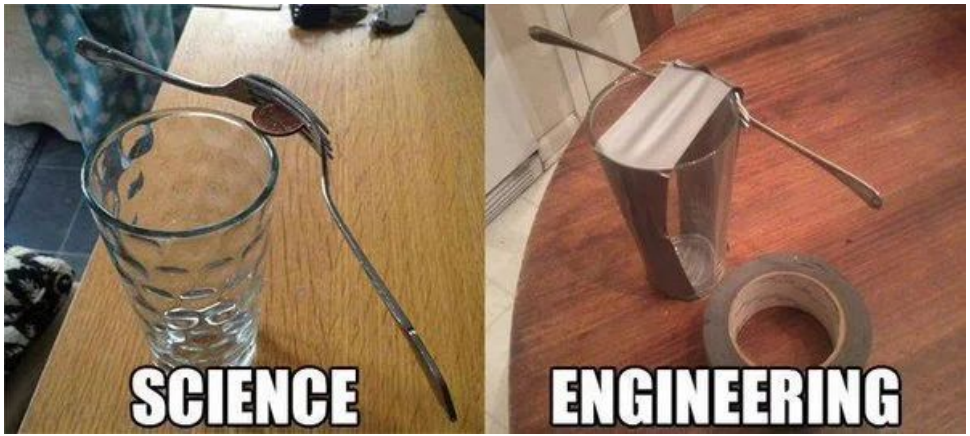
> *I used to see a lot of people complaining that model developers don't follow software engineering [practices]. At this point, I'm feeling more convinced that they don't follow software engineering [practices]—[not] because they're lazy, [but because software engineering practices are] contradictory to the agility of analysis and exploration.*

- Operationalizing Machine Learning: An Interview Study

Why Test?

No but like really

Engineer



We test ML software because…

- Solid bricks = solid building
- Code quality ensures maintainability
- The auto-grader is out to get you

We achieve this through…

- Pre-commit hooks
- Unit testing + code coverage
- Integration testing
- Version control + GitOps
- Code review
- Big ol’ CI jobs

Data Scientist

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

$$J = \frac{1}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i)^2$$

$$\frac{\partial J}{\partial a_0} = \frac{2}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i) \implies \frac{\partial J}{\partial a_0} = \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

$$\frac{\partial J}{\partial a_1} = \frac{2}{n}\sum_{i=1}^{n}(a_0 + a_1 \cdot x_i - y_i) \cdot x_i \implies \frac{\partial J}{\partial a_1} = \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) \cdot x_i$$

We test ML software because…

- Results that look right may not be right
- Good modeling is about tradeoffs
- The Staff Data Scientist is coming to question your math

We achieve this through…

- Experimentation
- Iteration
- Showing early prototypes to internal customers
- Stats. All teh statz.

C-Suite Type

We test ML software because…

- Outages and wrong results threaten our customers' trust in our product
- Customer distrust tanks the stock price
- The board is coming to get you in the Q3 earnings call

We achieve this through…

- Really thorough Root Cause Analyses after outages
- Holding direct reports to account

A Different Technical Problem

We are confusing the means (testing)

with the purpose (reliability)

Pursuing Reliability for Business Value

- Data Reliability Engineering
- Pre-prod / staging environments
- ML Observability

Data Reliability Engineering

[Data Reliability Engineering: A New Approach to Data Quality]

Data Council Austin 2022
Egor Gryaznov - Bigeye

1. Embrace Risk
2. Set standards
3. Reduce toil
4. Monitor Everything
5. Use automation
6. Control releases
7. Favor simplicity

Great Expectations is a tool to get started.

Did you know that Data Scientists spend most of their time cleaning the data?

So maybe we should actually orient our practices around that.

Pre-prod Environments

- Eliminate “but it worked in dev” or “test it in prod”
- Apply DRE and SRE principles to your pre-prod environment instead of building wacky integration test frameworks
- Prod-like environments with prod-like hardware and a prod-like stack

ML Observability

- Monitoring for performance and system stability
- Drift
- Feedback for training

Production Readiness Score

- Documentation
- Documentation
- Documentation again
- Also documentation
- Monitoring setup
- ML Observability setup
- Data Checks
- Governance compliance
- Ethical compliance

Model A

- ✅ Good tests
- ✅ Decent documentation
  - ✅ Readme
  - ✅ Confluence page for project
- ✅ High production stability
- ❌ No model retraining

PRS: 80%

Model B

- ❌ No tests
- ❌ Partial documentation
  - ✅ Readme
  - ❌ Confluence page for project
- ❌ High batch job failure rate
- ❌ No model retraining
- ❓ Really flashy, exciting demo that doesn't count for anything in the PRS

PRS: 8%

Where Traditional SE Testing Has A Place

- Internal libraries
- Feature engineering functions
- APIs

> “The reality is: every single thing I said that must or mustn't be done so far, I eventually found a situation where doing the opposite was the most appropriate choice.”
>
> -Eduardo Bonet, Staff MLOps @ Gitlab

Hot Take: Data Scientists should have an on call rotation

- Context
- Incentives
- Connection between quality and reliability

Velocity comes from good tools.

Quality comes from good incentives.

All Those Corporate Values

business people shaking hands because they just love their corporate values

Enter a negative prompt

Generate image

Better ML Reliability through…

- Honoring the working methods of our Data Scientists
- Prioritizing the customer
- Focusing on bigger picture inputs and outputs
- Aligning incentives

Extinguish the garbage fire of ML Testing by reorienting our thinking

*This is the uncool-est meme I've ever made

INTUIT mailchimp

Thank you!

Questions? Disagreements? Other ideas? Just wanna talk more?

Come to my office hours after lunch and afternoon keynote!

2pm in Classroom 108